Transform preprocessing for neural networks for object recogniition and localization with sonar

Billur Barshan
Birsel Ayrulu

Invited Paper

Transform pre-processing for neural networks for object recognition and localization with sonar

Billur Barshan* and Birsel Ayrulu

Department of Electrical Engineering, Bilkent University, Bilkent, TR-06800 Ankara, Turkey

ABSTRACT

We investigate the pre-processing of sonar signals prior to using neural networks for robust differentiation of commonly encountered features in indoor environments. Amplitude and time-of- ight measurement patterns acquired from a real sonar system are pre-processed using various techniques including wavelet transforms, Fourier and fractional Fourier transforms, and Kohonen's self-organizing feature map. Modular and non-modular neural network structures trained with the back-propagation and generating-shrinking algorithms are used to incorporate learning in the identi cation of parameter relations for target primitives. Networks trained with the generating-shrinking algorithm demonstrate better generalization and interpolation capability and faster convergence rate. The use of neural networks trained with the back-propagation algorithm, usually with fractional Fourier transform or wavelet pre-processing results in near perfect differentiation, around 85% correct range estimation and around 95% correct azimuth estimation, which would be satisfactory in a wide range of applications. Neural networks can differentiate more targets, employing only a single sensor node, with a higher correct differentiation percentage than achieved with previously reported methods employing multiple sensor nodes. The success of the neural network approach shows that the sonar signals do contain sufficient information to differentiate a considerable number of target types, but the previously reported methods are unable to resolve this identifying information. This work can nd application in areas where recognition of patterns hidden in sonar signals is required. Some examples are system control based on acoustic signal detection and identi cation, map building, navigation, obstacle avoidance, and target-tracking applications for mobile robots and other intelligent systems.

Keywords: arti cial neural networks, sonar sensing, input pre-processing, object recognition, position estimation, target differentiation, target localization, feature extraction, learning, fractional Fourier transform, discrete wavelet transform, acoustic signal processing

1. INTRODUCTION

Intelligent systems, especially those which interact with, or act upon their surroundings need the model of the environment in which they operate. They can obtain this model partly or entirely using one or more sensors and/or viewpoints. An important example of such systems is fully or partly autonomous mobile robots. For instance, considering typical indoor environments, a mobile robot must be able to differentiate planar walls, corners, edges, and cylinders for map-building, navigation, obstacle avoidance, and target-tracking applications.

Reliable differentiation is crucial for robust operation and is highly dependent on the mode(s) of sensing employed. Sonar sensing is one of the most useful and cost-effective modes of sensing. The fact that sonar sensors are light, robust and inexpensive devices has led to their widespread use in applications such as navigation of autonomous vehicles through unstructured environments,[1–3] map-building,[4–6] target-tracking,[7] and obstacle avoidance.[8] Although there are difficulties in the interpretation of sonar data due to poor angular resolution of sonar, multiple and higher-order re ections, and establishing correspondence between multiple echoes on different receivers,[9,10] these difficulties can be overcome by employing accurate physical models for the re ection of sonar. Sonar ranging systems commonly employ only the *time-of- ight* (TOF) information, recording the time elapsed between the transmission and reception of a pulse.[11] A review of work using this approach can be found in Refs.[12,13].

*E-mail: billur@ee.bilkent.edu.tr; phone: (90-312) 290-2161; fax: (90-312) 266-4192; www.ee.bilkent.edu.tr/∼ billur

Independent Component Analyses, Wavelets, and Neural Networks, Anthony J. Bell, Mladen V. Wickerhauser, Harold H. Szu, Editors, Proceedings of SPIE Vol. 5102 (2003) © 2003 SPIE · 0277-786X/03/$15.00

In the present paper, arti cial neural networks (ANNs) are used to process amplitude and TOF information with different pre-processing methods so as to reliably handle the target classi cation problem. The paper is organized as follows. Section 2 describes the sensing con guration used in this study and introduces the target primitives. In Section 3, multi-layer feed-forward ANNs are brie y reviewed. Two training algorithms, namely back-propagation and generating-shrinking algorithms, are described in Section 3.1. In Section 3.2, pre-processing techniques employed prior to ANNs are brie y described. In Section 3.3, various types of input signals to ANNs are proposed. In Section 4, the effect of these input signals and training algorithms on the performance of ANNs in target classi cation and localization are compared experimentally. In the last section, concluding remarks are made and directions for future work are discussed.

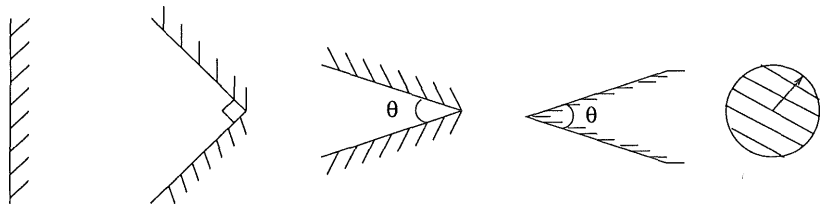

Figure 1. Horizontal cross sections of the target primitives/features differentiated in this study.

2. SONAR SENSING

The basic target types or features differentiated in this study are *plane*, *corner*, *acute corner*, *edge* and *cylinder* (Fig. 1). In particular, we have employed a planar target, a corner of $\theta_c = 90°$, an acute corner of $\theta_c = 60°$, an edge of $\theta_e = 90°$, and cylinders with radii $r_c = 2.5, 5.0$ and 7. 5cm, all made of wood. Detailed re ection models of these are provided in Ref. [14].

The most common sonar ranging system is based on *time-of- ight* (TOF) which is the time elapsed between the transmission and the reception of a pulse. In commonly used TOF systems, an echo is produced when the transmitted pulse encounters an object and a range measurement $r = ct_o/2$ is obtained (Fig. 2) by *simple thresholding*.[16] Here, t_o is the TOF and c is the speed of sound in air (at room temperature, $c = 343.3$ m/s.).

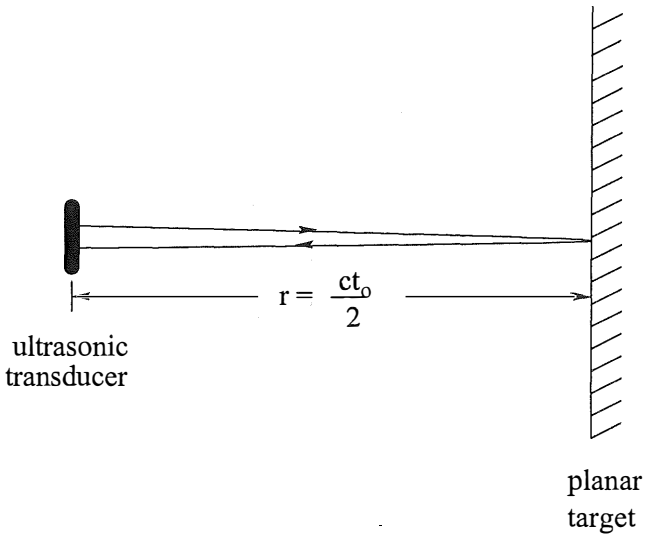

Figure 2. Re ection of ultrasonic echoes from a planar target.

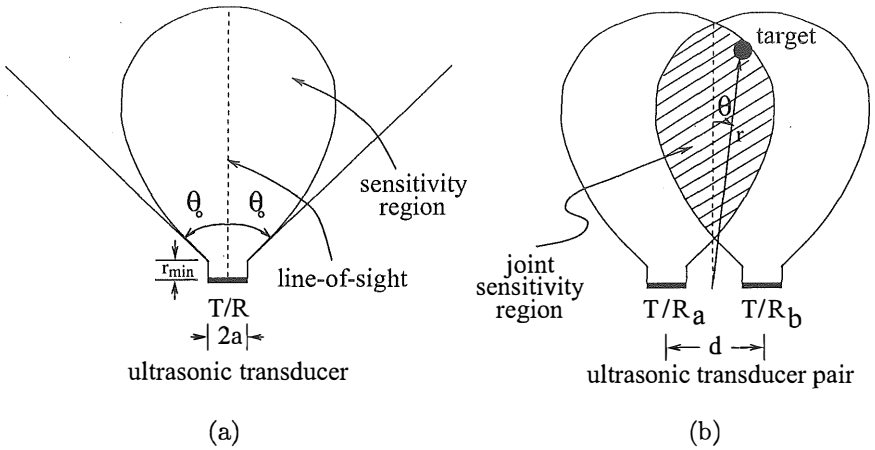

Figure 3. (a) Sensitivity region of an ultrasonic transducer. Sidelobes are not shown. (b) Joint sensitivity region of a pair of ultrasonic transducers. The intersection of the individual sensitivity regions serves as a reasonable approximation to the joint sensitivity region.

The major limitation of sonar sensors comes from their large beamwidth. Although these devices return accurate range data, they cannot provide direct information on the angular position of the object from which the re ection was obtained. The transducer can operate both as transmitter and receiver and detect echo signals re ected from targets within its *sensitivity region* (Figure 3(a)). Thus, with a single stationary transducer, it is not possible to estimate the azimuth of a target with better resolution than the angular resolution of the device which is approximately $2\theta_\circ$. This is usually not sufficient to differentiate more than a small number of target primitives.[17] The re ection point on the object can lie anywhere along a circular arc (as wide as the beamwidth) at the measured range. More generally, when one sensor transmits and another receives, both members of the sensor con guration can detect targets located within the *joint sensitivity region*, which is the overlap of the individual sensitivity regions (Figure 3(b)). In this case, the re ection point lies on the arc of an ellipse whose focal points are the transmitting and receiving transducers. The angular extent of these circular and elliptical arcs is determined by the sensitivity regions of the transducers. Improved target classi cation can be achieved by using multiple sensors and by employing both amplitude and TOF information. However, a major problem with using the amplitude information of sonar signals is that the amplitude is very sensitive to environmental conditions. For this reason, and also because the standard electronics used in practical work typically provide only TOF data, amplitude information is rarely used. Barshan and Kuc's early work on the use of amplitude information[17] has been extended to a variety of target types in Ref.[14] using both amplitude and TOF information. In the present paper, amplitude and TOF information from a pair of identical ultrasonic transducers a and b with center-to-center separation $d = 25$ cm is employed to improve the angular resolution.[15]

Panasonic transducers[18] with aperture radius $a = 0.65$ cm, resonance frequency $f_\circ = 40$ kHz, and beamwidth 108° are used in our experiments. The entire sensing unit is mounted on a small 6 V computer-controlled stepper motor with step size 1.8°. Data acquisition from the sonars is through a 12-bit 1 MHz PC A/D card. Starting at the transmit time, 10,000 samples of each echo signal are collected to record the peak amplitude and the TOF.

Amplitude and TOF patterns of the targets are collected in this manner at 25 different locations (r, θ) for each target, from $\theta = -20°$ to $\theta = 20°$ in 10° increments, and from $r = 35$ to 55 cm in 5 cm increments (Fig. 4). The target located at range r and azimuth θ is scanned by the rotating sensing unit for scan angles $-52° \leq \alpha \leq 52°$ with 1.8° increments (determined by the step size of the motor). The angle α is always measured with respect to $\theta = 0°$ as shown in Fig. 5.

At each step of the scan (for each value of α), four sonar echo signals are acquired. The echo signals are in the form of slightly skewed wave packets[13] (Fig. 6). In the gure, A_{aa}, A_{bb}, A_{ab}, and A_{ba} denote the peak values of the echo signals, and t_{aa}, t_{bb}, t_{ab}, and t_{ba} denote their TOF delays (extracted by simple thresholding). The rst subscript indicates the transmitting transducer, the second denotes the receiver. At each step of the

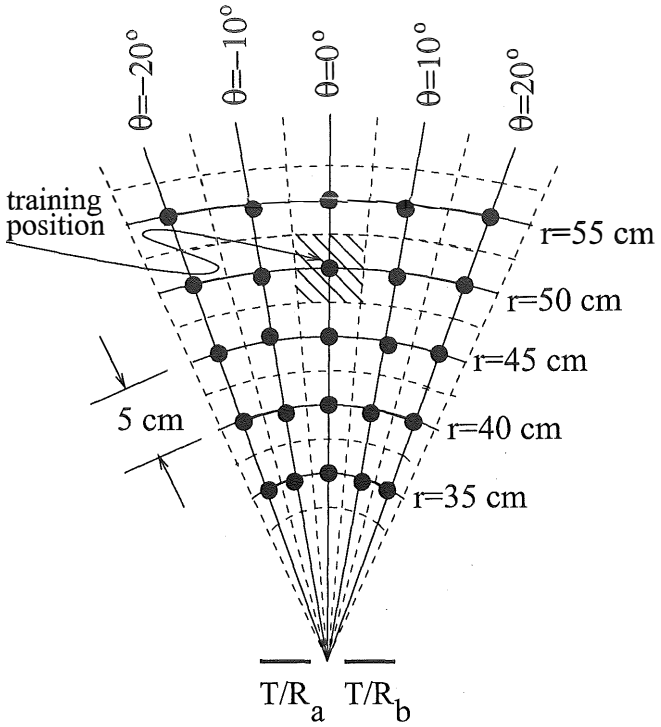

Figure 4. Discrete training locations. T/R_a and T/R_b denote the two transmitting/receiving transducers.

scan, only these eight amplitude and TOF values extracted from the four echo signals are recorded. For the given scan range and motor step size, 58 ($= \frac{2\times 52^\circ}{1.8^\circ}$) angular samples of each of the amplitude and TOF patterns $A_{aa}(\alpha)$, $A_{bb}(\alpha)$, $A_{ab}(\alpha)$, $A_{ba}(\alpha)$, $t_{aa}(\alpha)$, $t_{bb}(\alpha)$, $t_{ab}(\alpha)$, and $t_{ba}(\alpha)$ are acquired at each target location.

Since the cross terms $A_{ab}(\alpha)$ and $A_{ba}(\alpha)$ (or $t_{ab}(\alpha)$ and $t_{ba}(\alpha)$) should ideally be equal due to reciprocity, it is more representative to employ their average. Thus, 58 samples each of the following six functions are taken collectively as acoustic signatures embodying shape and position information of a given target:

$$A_{aa}(\alpha),\ A_{bb}(\alpha),\ \frac{A_{ab}(\alpha)+A_{ba}(\alpha)}{2},\ t_{aa}(\alpha),\ t_{bb}(\alpha),\ \text{and}\ \frac{t_{ab}(\alpha)+t_{ba}(\alpha)}{2}. \tag{1}$$

Scans are collected with 4-fold redundancy for each target primitive at each location, resulting in 700 (= 4-fold redundancy × 25 locations × 7 target types) sets of scans to be used for training. This set of 700 data is referred as the *training set* throughout this paper. This training set is used to design decision rules in statistical pattern recognition techniques and to train the ANNs.

In this study, three different *test sets* are acquired to evaluate and compare the different input pre-processing methods. For *test set I*, each target is placed in turn in each of the 25 training positions in Fig. 4. Again, scans are collected with 4-fold redundancy for each combination of target type and location, resulting in 700 sets of experimentally acquired scans. While collecting *test set II*, the targets are situated arbitrarily in the continuous estimation space and not necessarily con ned to one of the 25 training positions. The values of r, θ corresponding to these locations are randomly and uniformly generated in the range $r \in [32.5\text{ cm}, 57.5\text{ cm}]$ and $\theta \in [-25^\circ, 25^\circ]$. In collecting *test set III*, we employ targets not scanned during training which are slightly different in size, shape, or roughness than the targets used for training. These are two smooth cylinders of radii 4 cm and 10 cm, a cylinder of radius 7.5 cm and a plane both covered with blister packaging material, and a 60° smooth edge. The blister packaging material has a honeycomb pattern of uniformly distributed circular bubbles of diameter 1.0 cm and height 0.3 cm, with a center-to-center separation of 1.2 cm.

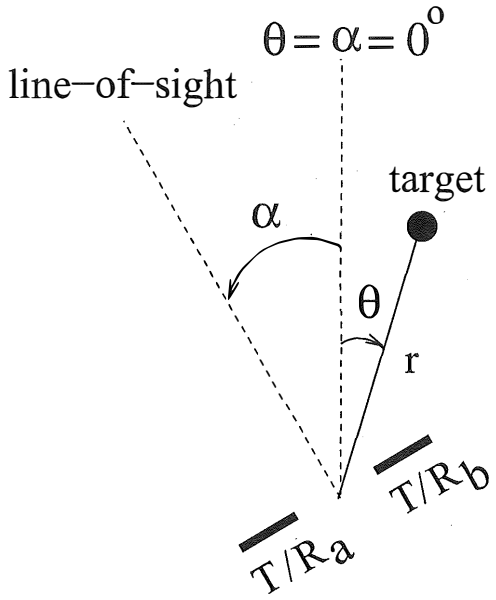

Figure 5. The scan angle α and the target azimuth θ.

We construct three alternative feature vector representations from the scans of Eqn. (1):
$\mathbf{x}_A : [\mathbf{A}_{aa},\ \mathbf{A}_{bb},\ \frac{\mathbf{A}_{ab}+\mathbf{A}_{ba}}{2},\ \mathbf{t}_{aa},\ \mathbf{t}_{bb},\ \frac{\mathbf{t}_{ab}+\mathbf{t}_{ba}}{2}]^T$
$\mathbf{x}_B : [\mathbf{A}_{aa}-\mathbf{A}_{ab},\ \mathbf{A}_{bb}-\mathbf{A}_{ba},\ \mathbf{t}_{aa}-\mathbf{t}_{ab},\ \mathbf{t}_{bb}-\mathbf{t}_{ba}]^T$
$\mathbf{x}_C : [(\mathbf{A}_{aa}-\mathbf{A}_{ab})(\mathbf{A}_{bb}-\mathbf{A}_{ba}), (\mathbf{A}_{aa}-\mathbf{A}_{ab})+(\mathbf{A}_{bb}-\mathbf{A}_{ba}), (\mathbf{t}_{aa}-\mathbf{t}_{ab})(\mathbf{t}_{bb}-\mathbf{t}_{ba}), (\mathbf{t}_{aa}-\mathbf{t}_{ab})+(\mathbf{t}_{bb}-\mathbf{t}_{ba})]^T$
Here, $\mathbf{A}_{aa}$ denotes the row vector representing the samples of $A_{aa}(\alpha)$ at the 58 scan angles. The rst feature vector $\mathbf{x}_A$ is taken as the original form of the scans, except for averaging the cross terms. The choice of the second feature vector $\mathbf{x}_B$ has been motivated by the target differentiation algorithm in Ref.[14].The third feature vector $\mathbf{x}_C$ is motivated by the differential terms which are used to assign belief values to the target types in Dempster-Shafer evidential reasoning and majority voting.[14] Note that the dimensionalities d of these vector representations are 348 (= 6 × 58), 232 (= 4 × 58), and 232, respectively.

3. ARTIFICIAL NEURAL NETWORKS

ANNs have been widely used in areas such as target detection and classi cation,[19] speech processing,[20] system identi cation,[21] control theory,[22] medical applications,[23] and character recognition.[24] In this study, ANNs are employed to identify and resolve parameter relations embedded in the characteristics of sonar echo returns from all seven target types considered, for their differentiation and localization in a robust manner in real time. ANNs consist of an input layer, one or more hidden layers to extract progressively more meaningful features, and a single output layer, each comprised of a number of units called neurons. The model of each neuron includes a smooth nonlinearity, here a sigmoid function of the form $\varphi(v) = (1+e^{-v})^{-1}$. Due to the presence of distributed nonlinearity and a high degree of connectivity, theoretical analysis of ANNs is difficult. These networks are trained to compute the boundaries of decision regions in the form of connection weights and biases by using training algorithms. Performance of ANNs is affected by the choice of parameters related to the network

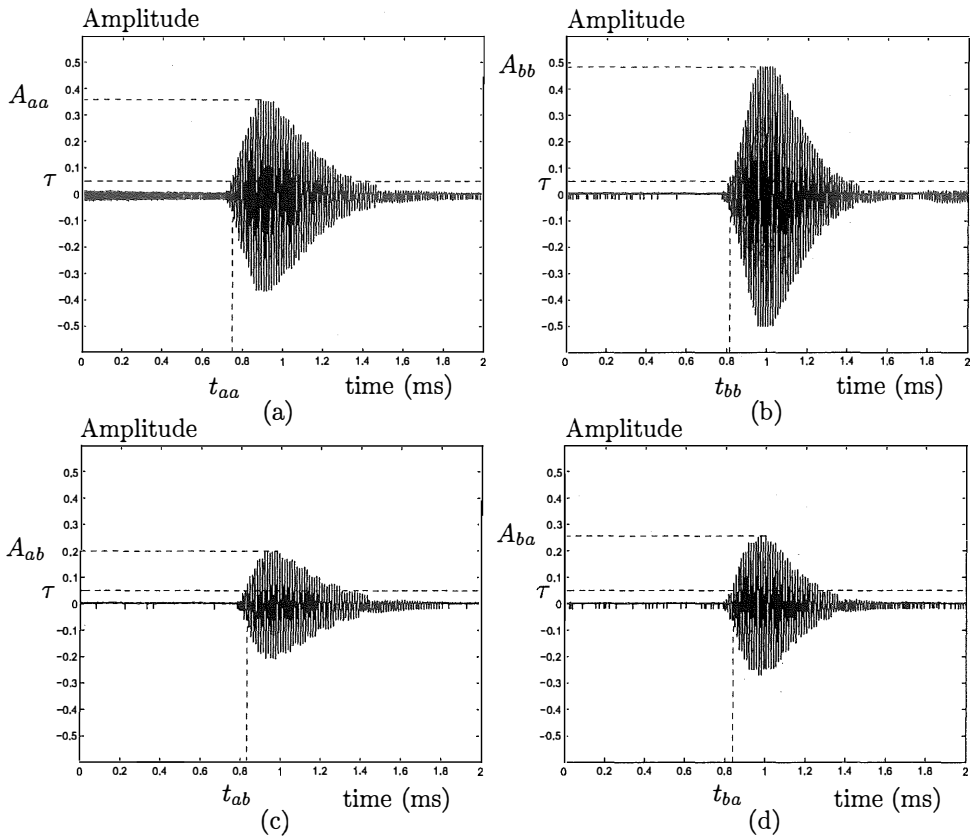

Figure 6. Real sonar signals obtained from a planar target when (a) transducer a transmits and transducer a receives (b) transducer b transmits and b receives (c) transducer a transmits and b receives (d) transducer b transmits and a receives.

structure, training algorithm, and input signals, as well as parameter initialization.[25] In this study, two training algorithms are employed, namely, back-propagation (BP) and generating-shrinking (GS) algorithms.

3.1. Training Algorithms

3.1.1. Back-Propagation (BP) Algorithm

With the BP algorithm, a set of *training patterns* is presented to the network and the error between the resulting signal at the output and the desired signal is minimized with a gradient-descent procedure. The two adjustment parameters of the algorithm, namely the learning rate and the momentum constant[26] are chosen to be 0.01 and 0.9, respectively, and training with the BP algorithm is stopped either when the average error is reduced to 0.001 or if a maximum of 10,000 epochs is reached, whichever occurs earlier. The second case occurs very rarely. The number of hidden-layer neurons is determined by enlarging.[27]

3.1.2. Generating-Shrinking (GS) Algorithm

The GS algorithm rst builds and then shrinks or prunes a feed-forward neural network, offering fast convergence rates and 100% correct classi cation on the training set.[28] The network used in Ref. [28] consists of two hidden layers with equal numbers of neurons, initially set equal to the number of training patterns. Pre-determined initial connection weights are assigned, with the consequence that the generalization behavior of the network is analytically known. Then, the hidden layers are pruned while preserving 100% correct classi cation on the training set. Only one output neuron takes the value one (the winning neuron) and the remaining output neurons take the value zero. At the input layer, a pre- xed reference number $n_r \in (0, \infty)$ is used as an additional input to control the generalization capability of the network. The algorithm achieves scale-invariant generalization

behavior as n_r approaches zero, and behaves like a nearest-neighborhood classi er as it tends to in nity. We employ the relatively small value $n_r = 0.01$ in order to enhance scale invariance. A comparison with the BP algorithm[28] indicates that the GS algorithm does not have the convergence problems of the BP algorithm and has several hundred times faster convergence rate and improved generalization capability.

3.2. Pre-processing of the input signals

The results obtained depend on which form the observed signals are presented to the ANNs. Therefore, we have considered several different pre-processing techniques.

3.2.1. Ordinary Fourier transform

The Fourier transform is widely used in signal processing to study the spectral behavior of a signal. The discrete Fourier transform (DFT) of a signal $f(n)$ is de ned as:

$$F(k) = \mathcal{F}\{f(n)\} \triangleq \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-i2\pi nk/N} \tag{2}$$

where N is the length of the discrete signal $f(n)$.

3.2.2. Fractional Fourier transform

The ath-order fractional Fourier transform is a generalization of the ordinary Fourier transform such that the 1st-order fractional Fourier transform is the ordinary Fourier transform and the 0th-order fractional Fourier transform corresponds to the function itself.[29] The transform has been studied extensively since the early 1990s with applications in wave propagation and optics,[30–33] time-frequency analysis, pattern recognition, and digital signal[34,35] and image processing.[36,37] Most applications are based on replacing the ordinary Fourier transform with the fractional transform. Since the latter has an additional degree of freedom (the order parameter a), it is often possible to generalize and improve upon previous results. The ath-order fractional Fourier transform $f_a(u)$ of $f(u)$ is de ned for $0 < |a| < 2$ as[35]

$$\begin{aligned} f_a(u) &\triangleq \int_{-\infty}^{\infty} A_\phi \exp\left[i\pi(u^2 \cot\phi - 2uu' \csc\phi + u'^2 \cot\phi)\right] f(u')du' \\ \text{where} \quad A_\phi &= \frac{\exp[-i(\pi \text{sgn}(\phi)/4 - \phi/2)]}{|\sin\phi|^{1/2}} \quad \text{and} \quad \phi = \frac{a\pi}{2} \end{aligned} \tag{3}$$

The $f_a(u)$ approaches $f(u)$ and $f(-u)$ as a approaches 0 and ± 2, respectively, and is de ned as such at these values. The fractional Fourier transform reduces to the ordinary Fourier transform when $a = 1$. The transform is linear and index additive: the a_1th-order transform of the a_2th-order transform is equal to the $(a_1 + a_2)$th-order transform. Digital implementation of the fractional Fourier transform is as efficient as that of the ordinary Fourier transform; it can also be computed in the order of $N \log N$ time.[29]

With a similar notation as in the case of DFT, the ath-order discrete fractional Fourier transform (DFRT) of $\mathbf{f}$, denoted $\mathbf{f_a}$, can be expressed as $\mathbf{f_a} = \mathbf{F^a f}$ where $\mathbf{F^a}$ is the $N \times N$ DFRT matrix which corresponds to the ath power of the ordinary DFT matrix $\mathbf{F}$ and $\mathbf{f}$ is an $N \times 1$ column vector.[38] However, we note that there are certain subtleties and ambiguities in de ning the power function.[38]

3.2.3. Hartley transform

Hartley transform[39] is a widely-used technique in signal processing applications such as image compression[40] and adaptive ltering.[41] The discrete Hartley transform (DHT) of $f(n)$ is de ned as:

$$H(k) = \mathcal{H}\{f(n)\} \triangleq \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) \text{ cas}\left(\frac{2\pi}{N} nk\right) \tag{4}$$

where $\text{cas}(x) \triangleq \cos(x) + \sin(x)$. If the DFT of a signal $f(n)$ is expressed as $F(k) = F_R(k) - iF_I(k)$, then its DHT is given by $H(k) = F_R(k) + F_I(k)$. The DHT can also be represented in matrix notation as $\mathbf{h_1} = \mathbf{Hf}$, where $\mathbf{H}$ is the $N \times N$ DHT matrix, and $\mathbf{h_1}$ is the DHT of $\mathbf{f}$.

3.2.4. Wavelet transform

We describe the discrete wavelet transform (DWT)[42] by referring to Fig. 7, where the operations performed on the input signal $f(n)$ of length N are shown as a block diagram. $h(-n)$ and $g(-n)$ are referred to as the *scaling filter* and the *wavelet filter*, respectively, where $g(n) \triangleq (-1)^n h(M-n-1)$. Mathematically,

$$\begin{aligned} c_j(k) &= \sum_m h(m-2k)c_{j+1}(m) \\ d_j(k) &= \sum_m g(m-2k)c_{j+1}(m) \qquad k = 0, 1, \ldots, (2^j N - 1) \quad \text{and} \quad j = -1, -2, \ldots \end{aligned} \tag{5}$$

where for $j = -1$ we associate $c_0(.)$ with $f(.)$ and these equations describe the left part of Fig. 7. When $j = -2$, they describe the right part of the same figure. More generally, these equations allow us to obtain the coefficients at scale j from the coefficients at scale $j+1$. We have employed the value $M = 23$ and the scaling filter whose coefficients $h(n)$ are given below:

$$\begin{array}{lllllllllll} h(n) = [& -0.002 & -0.003 & 0.006 & 0.006 & -0.013 & 0.012 & -0.030 & 0.023 & -0.078 & -0.035 & 0.307 \\ & 0.542 & 0.307 & -0.035 & -0.078 & 0.023 & -0.030 & 0.012 & -0.013 & 0.006 & 0.006 & -0.003 & -0.002 \;] \end{array}$$

for $n = 0, \ldots, M-1$. This filter is known as the Lamaire wavelet.[43] After down-sampling, the total number of samples in the concatenation of c_j and d_j is equal to the number of samples of c_{j+1}. In principle, the concatenation of c_j and d_j for any resolution level $j = -1, -2, \ldots$ can be used as an input to the neural network. However, values of j further than -2 were not found to be advantageous in our implementations as discussed later.

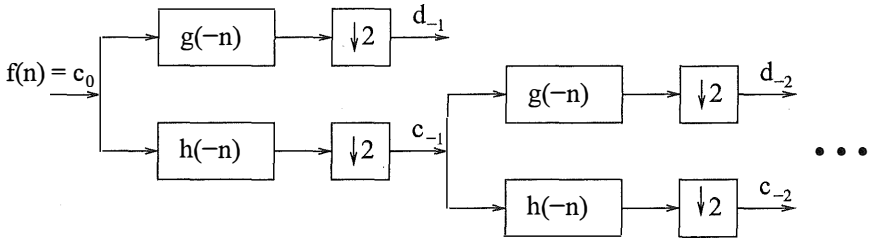

Figure 7. Block diagram of the DWT. The square boxes represent down-sampling.

3.2.5. Self-organizing feature map

Self-organizing ANNs are generated by unsupervised learning algorithms that have the ability to form an internal representation of the network, modeling the underlying structure of the input data. These networks are commonly used to solve the scale-variance problem encountered in supervised learning. However, it is not recommended to use them by themselves for pattern classification or other decision-making processes.[27] Best results are achieved with these networks when they are used as feature extractors prior to a linear classifier or a supervised learning procedure. The most commonly used algorithm for generating self-organizing ANNs is Kohonen's self-organizing feature-mapping (KSOFM) algorithm.[44] In this algorithm, weights are adjusted from the input layer towards the output layer where the output neurons are interconnected with local connections. These output neurons are geometrically organized in one, two, three, or even higher dimensions. This algorithm can be summarized as follows: (i) initialize the weights randomly, (ii) present new input from the training set, (iii) find the winning neuron at the output layer, (iv) select the neighborhood of this output neuron, (v) update weights from input towards selected output neurons, (vi) continue with the second step until no considerable changes in the weights occur (see Ref.[27] for further details).

3.3. Input signals

In this work, many different signal representations are considered as alternative inputs to the ANNs. In addition to the pre-processing methods discussed, different combinations of the amplitude and TOF patterns are also

considered. Speci cally, we employed the following 30 alternative inputs to the ANNs:
I_1 : samples of $A_{aa}(\alpha)$, $A_{bb}(\alpha)$, $\frac{A_{ab}(\alpha)+A_{ba}(\alpha)}{2}$, $t_{aa}(\alpha)$, $t_{bb}(\alpha)$, and $\frac{t_{ab}(\alpha)+t_{ba}(\alpha)}{2}$
I_2 : samples of $A_{aa}(\alpha) - A_{ab}(\alpha)$, $A_{bb}(\alpha) - A_{ba}(\alpha)$, $t_{aa}(\alpha) - t_{ab}(\alpha)$, and $t_{bb}(\alpha) - t_{ba}(\alpha)$
I_3 : samples of $[A_{aa}(\alpha) - A_{ab}(\alpha)][A_{bb}(\alpha) - A_{ba}(\alpha)]$, $[A_{aa}(\alpha) - A_{ab}(\alpha)] + [A_{bb}(\alpha) - A_{ba}(\alpha)]$, $[t_{aa}(\alpha) - t_{ab}(\alpha)][t_{bb}(\alpha) - t_{ba}(\alpha)]$, and $[t_{aa}(\alpha) - t_{ab}(\alpha)] + [t_{bb}(\alpha) - t_{ba}(\alpha)]$
$I_4 - I_{12}$: DFT of I_1, I_2, I_3, its low-frequency component (LFC), and its magnitude ($\mathcal{F}(I_i)$, LFC($\mathcal{F}(I_i)$), |LFC($\mathcal{F}(I_i)$)|, $i = 1, 2, 3$)
$I_{13} - I_{15}$: DFRT of I_1, I_2, I_3 at different orders ($\mathcal{F}^a(I_i)$, $i = 1, 2, 3$)
$I_{16} - I_{18}$: DHT of I_1, I_2, I_3 ($\mathcal{H}(I_i)$, $i = 1, 2, 3$)
$I_{19} - I_{27}$: DWT of I_1, I_2, I_3 and its low-frequency components at different resolutions (DWT(I_i), LFC(DWT(I_i))$_1$, LFC(DWT(I_i))$_2$, $i = 1, 2, 3$)
$I_{28} - I_{30}$: features extracted by using KSOFM (KSOFM(I_i), $i = 1, 2, 3$)

The sampled sequences I_1, I_2, I_3 correspond to the feature vectors $\mathbf{x}_A, \mathbf{x}_B$, and $\mathbf{x}_C$ de ned in Section rf:sonar. Here, they have been used both in their raw form and after taking their discrete ordinary and fractional Fourier, Hartley, and wavelet transforms, as well as after feature extraction by KSOFM. The transforms are performed on the six parts of I_1 and the four parts of I_2 and I_3 separately.

DWTs of each signal at different resolution levels j have been considered. Initially, DWT of each signal at resolution level $j = -1$ is used as the input: DWT(I_i), $i = 1, 2, 3$. Secondly, only the low-frequency component of the DWT, the c_{-1}'s, are employed: LFC(DWT(I_i))$_1$. Finally, the low-frequency component of DWT at resolution $j = -2$, the c_{-2}'s, are used: LFC(DWT(I_i))$_2$. Use of the low-frequency components helps eliminate high-frequency noise. However, more negative values of j, which correspond to fewer samples of c_j and d_j, and thus lower resolutions, lead to deterioration in the performance of the network beyond $j = -2$. The value $j = -2$ corresponds to the frequency-domain information between 0 and $\frac{\pi}{4}$ of the original patterns. To make a fair comparison, the low-frequency component of the DFT, LFC($\mathcal{F}(I_i)$), corresponding to the same frequency interval as LFC(DWT(I_i))$_2$ is also considered. We also employed the magnitude of the low-frequency component of the DFT, |LFC($\mathcal{F}(I_i)$)|. The ath-order DFRTs of the three input signal representations, for values of a varying from 0.05 to 0.95 with 0.05 increments have been considered. The features extracted by using KSOFM are used both prior to ANNs trained with the two training algorithms and prior to linear classi ers designed by using a least-squares approach.

Initially, a single integrated ANN is trained by using the BP algorithm to both classify and localize the targets for each of the above input signals. Next, modular network structures for each type of input signal have been considered in which three separate networks for target type, range, and azimuth, each trained with the BP algorithm, are employed. Neural networks using the same input signal representations are also trained with the GS algorithm. This algorithm can only be applied for target type classi cation since here only one output neuron takes the value one (the winning neuron) and the others are zero. For this reason, range and azimuth estimation cannot be made with this approach.[12]

4. RESULTS

As already mentioned, ANNs trained with the BP algorithm estimate the target type, range, and azimuth, whereas those trained with the GS algorithm determine only the target type. For non-modular and modular networks trained with the BP algorithm, the resulting average percentages over all target types for correct type classi cation, correct range and azimuth estimation are given in Table 1. (A range or azimuth estimate is considered correct if it is within an error tolerance of ϵ_r of the actual range or ϵ_θ of the actual azimuth.) In this 3-panel table, the numbers before the parentheses are for non-modular networks, whereas the numbers in the parentheses are for modular networks. For the DFRT, results are given for the corresponding optimal value of a.[45] For test set I, the highest average percentage of correct classi cation of 100% is obtained with the input signal $\mathcal{F}^a(I_1)$ for non-modular networks, and 99% with LFC(DWT(I_1))$_2$ for modular networks. For non-modular networks, the highest average percentages of correct range estimation lie in the range 79–97% as the error tolerance ϵ_r varies between 0.125–10 cm. The optimal pre-processing method is one of $I_3, \mathcal{F}^a(I_1)$, or $\mathcal{F}(I_1)$. The highest average percentages of correct azimuth estimation lie in the range 93–100% as the error tolerance ϵ_θ varies between 0.25–20°. The optimal pre-processing method is usually $\mathcal{F}^a(I_1)$ or LFC(DWT(I_1))$_2$.

test set I: input to ANN	% of correct classif.	% of correct r estimation, error tolerance ϵ_r ±0.125 cm	±1 cm	± 5 cm	±10 cm	% of correct θ estimation, error tolerance ϵ_θ ±0.25°	±2°	±10°	±20°
I_1	88(88)	30(33)	41(46)	63(70)	86(87)	65(65)	76(72)	87(84)	97(97)
I_2	95(95)	74(73)	77(88)	87(93)	93(96)	89(95)	92(96)	95(97)	97(99)
I_3	86(88)	79(73)	82(75)	89(83)	94(91)	83(87)	89(91)	95(95)	97(98)
$\mathcal{F}(I_1)$	97(98)	64(72)	69(73)	86(87)	96(95)	86(94)	93(96)	96(98)	100(100)
LFC($\mathcal{F}(I_1)$)	96(97)	56(70)	64(73)	86(88)	95(97)	84(92)	90(96)	96(96)	100(99)
|LFC($\mathcal{F}(I_1)$)|	88(86)	28(45)	35(52)	68(77)	88(93)	65(55)	70(59)	86(79)	95(90)
$\mathcal{F}(I_2)$	93(89)	59(60)	64(65)	79(78)	89(90)	76(73)	81(86)	88(91)	93(96)
LFC($\mathcal{F}(I_2)$)	99(95)	63(68)	72(74)	85(86)	94(92)	91(89)	93(91)	96(96)	99(98)
|LFC($\mathcal{F}(I_2)$)|	86(95)	35(54)	42(60)	73(80)	96(94)	39(56)	50(65)	71(86)	86(95)
$\mathcal{F}(I_3)$	86(90)	54(62)	61(65)	77(77)	89(89)	70(77)	76(82)	85(88)	94(94)
LFC($\mathcal{F}(I_3)$)	91(85)	60(60)	68(65)	82(78)	92(90)	77(78)	81(83)	88(89)	96(96)
|LFC($\mathcal{F}(I_3)$)|	74(82)	34(41)	42(49)	65(72)	85(90)	30(53)	39(60)	62(78)	83(90)
$\mathcal{F}^a(I_1)$	100(96)	75(62)	79(66)	89(86)	97(96)	93(76)	96(79)	97(92)	100(99)
$\mathcal{F}^a(I_2)$	98(98)	67(68)	71(76)	83(87)	92(95)	80(86)	84(89)	90(95)	96(98)
$\mathcal{F}^a(I_3)$	90(93)	61(59)	68(62)	83(80)	92(90)	76(75)	82(79)	88(88)	95(94)
$\mathcal{H}(I_1)$	99(97)	59(54)	68(60)	85(81)	94(94)	84(84)	89(87)	95(95)	98(99)
$\mathcal{H}(I_2)$	98(97)	67(62)	72(68)	85(80)	93(90)	80(84)	85(86)	91(93)	96(99)
$\mathcal{H}(I_3)$	87(81)	59(46)	66(51)	80(69)	90(89)	73(79)	80(84)	89(90)	95(95)
DWT(I_1)	82(74)	15(21)	30(27)	59(59)	80(82)	46(51)	58(63)	77(80)	94(94)
LFC(DWT(I_1))$_1$	85(98)	18(21)	28(33)	58(59)	82(79)	54(49)	65(62)	80(79)	95(94)
LFC(DWT(I_1))$_2$	98(99)	71(80)	76(82)	87(91)	95(96)	90(92)	93(93)	97(98)	100(100)
DWT(I_2)	92(96)	63(64)	69(69)	84(82)	93(92)	85(87)	88(90)	93(94)	96(96)
LFC(DWT(I_2))$_1$	95(97)	65(66)	70(71)	84(84)	94(91)	87(88)	90(90)	94(94)	97(96)
LFC(DWT(I_2))$_2$	89(84)	28(32)	34(44)	58(68)	84(88)	58(53)	68(61)	86(80)	95(92)
DWT(I_3)	86(89)	58(58)	62(62)	76(76)	93(89)	85(76)	88(80)	93(88)	96(94)
LFC(DWT(I_3))$_1$	82(91)	56(61)	60(66)	75(78)	89(87)	73(79)	77(83)	86(89)	93(94)
LFC(DWT(I_3))$_2$	83(79)	29(33)	37(44)	63(69)	83(88)	53(41)	65(52)	78(75)	87(89)
KSOFM(I_1)	75(74)	17(14)	25(23)	49(46)	80(72)	64(61)	67(64)	81(79)	90(89)
KSOFM(I_2)	78(76)	22(19)	28(28)	59(57)	88(81)	69(66)	73(71)	86(85)	92(93)
KSOFM(I_3)	66(63)	24(21)	30(31)	57(55)	84(81)	51(49)	54(51)	78(75)	89(87)

Table 1. Average percentages of correct classi cation, range (r) and azimuth (θ) estimation for ANNs trained with the BP algorithm.

For modular networks, the highest average percentage of correct range estimation varies between 80–96% as ϵ_r varies between 0.125–10 cm. This is obtained with either $I_2, \mathcal{F}(I_1)$, or LFC(DWT(I_1))$_2$. The highest average percentage of correct azimuth estimation varies between 95–100% as the error tolerance level ϵ_θ varies between 0.25°–20°. The optimal pre-processing method is one of $I_2, \mathcal{F}(I_1)$, LFC($\mathcal{F}(I_1)$), or LFC(DWT(I_1))$_2$.

In general, straightforward use of DWT pre-processing does not offer any improvements with respect to no pre-processing. However, the low-frequency part of the DWT does offer better performance, with the resolution level ($j = -1$ or $j = -2$) to be used depending on whether we use I_1, I_2, or I_3. Employing the low-frequency part of the Fourier transform gives better classi cation and estimation performance than employing the whole Fourier transform for the input signals I_2 and I_3, while giving comparable results for I_1. (The ordinary Fourier transform can be considered as a special case of the DFRT.)

For test set II (Table 1-panel 2), the maximum correct target classi cation percentages of 100% (non-modular) and 99% (modular) are obtained when the input signals $\mathcal{F}^a(I_1)$ and LFC(DWT(I_1)$_2$ are used, respectively. These values are the same as those achieved with test set I. However, the percentages for correct range and azimuth estimates are generally 3–16% and 0–30% lower than test set I, respectively. Noting that the networks are trained only at 25 locations and at grid spacings of 5 cm and 10°, it can be concluded from the percentage of correct range and azimuth estimates obtained at error tolerances of $|\epsilon_r| = 0.125$ cm and 1 cm and $|\epsilon_\theta| = 0.25°$ and 2°, that the networks demonstrate the ability to interpolate between the training grid locations. Thus, the neural network maintains a certain spatial continuity between its input and output and does not haphazardly map positions which are not drawn from the 25 locations of Fig. 4. The correct target type percentages are just as good (99–100%) and the accuracy of the range/azimuth estimates would be acceptable for most applications. If better estimates are required, this can be achieved by reducing the training grid spacing in Fig. 4. Finally, we add that the results for the modular networks are slightly better than those for the non-modular networks. Furthermore, use of modular networks has the additional advantage that one can independently optimize the pre-processing method and the parameters.

test set II: input to ANN	% of correct classif.	% of correct r estimation, error tolerance ϵ_r				% of correct θ estimation, error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	$\pm 0.25^\circ$	$\pm 2^\circ$	$\pm 10^\circ$	$\pm 20^\circ$
I_1	88(88)	17(18)	32(30)	55(56)	78(83)	37(38)	47(47)	75(74)	91(94)
I_2	90(93)	59(60)	63(69)	78(83)	88(88)	70(71)	75(76)	92(97)	94(98)
I_3	58(59)	63(60)	63(62)	76(76)	83(85)	66(69)	74(73)	93(93)	94(97)
$\mathcal{F}(I_1)$	96(98)	53(57)	54(57)	81(75)	91(88)	69(72)	77(77)	89(98)	98(98)
LFC($\mathcal{F}(I_1)$)	96(97)	52(59)	58(62)	82(83)	89(89)	69(69)	75(74)	83(83)	98(98)
|LFC($\mathcal{F}(I_1)$)|	86(82)	20(37)	28(45)	64(72)	86(88)	57(53)	66(59)	78(74)	88(88)
$\mathcal{F}(I_2)$	89(92)	52(51)	53(52)	67(68)	80(80)	60(59)	65(68)	81(92)	83(95)
LFC($\mathcal{F}(I_2)$)	98(95)	54(56)	57(58)	74(70)	83(80)	69(69)	72(73)	95(90)	97(92)
|LFC($\mathcal{F}(I_2)$)|	83(90)	21(42)	31(50)	64(74)	86(92)	39(51)	49(60)	71(75)	81(87)
$\mathcal{F}(I_3)$	84(87)	48(51)	52(53)	65(68)	77(80)	57(60)	63(65)	82(84)	89(86)
LFC($\mathcal{F}(I_3)$)	90(85)	56(53)	56(54)	74(73)	85(85)	61(62)	65(67)	87(86)	91(91)
|LFC($\mathcal{F}(I_3)$)|	74(81)	25(36)	34(43)	57(60)	80(86)	30(48)	39(56)	62(78)	81(87)
$\mathcal{F}^a(I_1)$	100(96)	59(53)	60(55)	79(79)	89(88)	70(63)	75(68)	97(97)	100(99)
$\mathcal{F}^a(I_2)$	92(92)	55(56)	55(59)	67(71)	78(83)	62(65)	67(68)	85(91)	90(92)
$\mathcal{F}^a(I_3)$	83(85)	53(52)	53(54)	72(71)	81(79)	61(60)	70(65)	85(80)	89(88)
$\mathcal{H}(I_1)$	92(96)	52(51)	55(54)	76(77)	87(89)	68(67)	74(73)	93(95)	96(99)
$\mathcal{H}(I_2)$	93(95)	55(52)	58(52)	71(68)	83(82)	62(66)	68(71)	86(94)	90(96)
$\mathcal{H}(I_3)$	77(79)	50(44)	51(45)	72(66)	83(83)	60(61)	65(68)	81(86)	87(87)
DWT(I_1)	82(74)	12(14)	24(20)	50(53)	76(79)	26(29)	37(38)	64(64)	87(89)
LFC(DWT(I_1))$_1$	85(98)	11(13)	22(22)	50(53)	75(75)	33(31)	41(43)	70(71)	87(91)
LFC(DWT(I_1))$_2$	98(99)	60(64)	60(64)	76(79)	91(89)	71(72)	77(77)	96(94)	96(95)
DWT(I_2)	92(93)	53(54)	53(57)	72(71)	85(81)	65(66)	67(69)	87(90)	92(92)
LFC(DWT(I_2))$_1$	91(94)	53(56)	53(56)	70(71)	80(80)	68(66)	72(70)	91(88)	91(90)
LFC(DWT(I_2))$_2$	86(80)	16(20)	28(29)	51(60)	80(79)	33(28)	40(34)	74(72)	86(88)
DWT(I_3)	82(85)	49(51)	53(52)	68(67)	78(81)	57(59)	63(65)	85(85)	87(88)
LFC(DWT(I_3))$_1$	80(86)	52(54)	52(54)	68(65)	80(77)	60(62)	67(68)	85(86)	88(90)
LFC(DWT(I_3))$_2$	80(78)	21(20)	30(32)	60(62)	81(83)	28(23)	38(31)	65(66)	84(84)
KSOFM(I_1)	75(73)	12(10)	19(18)	45(41)	77(69)	38(34)	40(37)	75(69)	88(86)
KSOFM(I_2)	78(76)	19(16)	23(21)	53(52)	82(78)	39(38)	45(42)	77(76)	88(87)
KSOFM(I_3)	65(61)	21(19)	26(25)	51(51)	78(73)	29(27)	34(33)	69(67)	81(80)

test set III: input to ANN	% of correct classif.	% of correct r estimation, error tolerance ϵ_r				% of correct θ estimation, error tolerance ϵ_θ			
		± 0.125 cm	± 1 cm	± 5 cm	± 10 cm	$\pm 0.25^\circ$	$\pm 2^\circ$	$\pm 10^\circ$	$\pm 20^\circ$
I_1	85(73)	18(21)	28(32)	49(55)	74(76)	35(40)	45(45)	61(56)	80(72)
I_2	78(80)	59(60)	59(65)	72(77)	83(84)	68(70)	73(75)	75(76)	76(80)
I_3	57(54)	60(59)	60(59)	69(69)	80(80)	64(68)	72(75)	73(78)	74(79)
$\mathcal{F}(I_1)$	77(74)	56(58)	57(58)	73(76)	87(83)	68(69)	78(77)	81(77)	81(77)
LFC($\mathcal{F}(I_1)$)	77(78)	52(59)	56(59)	73(73)	82(85)	69(68)	75(74)	83(83)	85(85)
|LFC($\mathcal{F}(I_1)$)|	68(68)	19(31)	25(36)	57(62)	81(82)	55(50)	60(56)	73(66)	78(74)
$\mathcal{F}(I_2)$	79(76)	50(53)	52(54)	59(67)	76(80)	54(61)	62(70)	71(77)	76(82)
LFC($\mathcal{F}(I_2)$)	84(81)	54(56)	57(58)	73(70)	83(79)	69(68)	72(73)	85(80)	87(86)
|LFC($\mathcal{F}(I_2)$)|	63(70)	21(35)	30(41)	60(67)	84(88)	34(42)	40(52)	68(65)	80(83)
$\mathcal{F}(I_3)$	74(76)	47(52)	48(52)	63(62)	76(75)	62(63)	69(72)	78(76)	81(79)
LFC($\mathcal{F}(I_3)$)	77(74)	52(53)	55(54)	68(66)	79(76)	61(62)	65(67)	71(73)	85(87)
|LFC($\mathcal{F}(I_3)$)|	65(70)	23(30)	31(38)	57(60)	79(82)	29(40)	36(48)	53(72)	72(79)
$\mathcal{F}^a(I_1)$	83(89)	61(55)	63(55)	77(72)	90(82)	67(67)	71(70)	71(80)	71(83)
$\mathcal{F}^a(I_2)$	81(79)	55(56)	56(57)	68(70)	79(79)	64(65)	70(72)	71(73)	73(77)
$\mathcal{F}^a(I_3)$	77(79)	52(53)	53(53)	65(64)	76(72)	62(62)	69(67)	73(75)	77(80)
$\mathcal{H}(I_1)$	89(87)	53(51)	54(52)	71(70)	79(80)	69(72)	76(76)	80(83)	80(83)
$\mathcal{H}(I_2)$	80(81)	56(52)	58(53)	72(66)	85(79)	65(66)	70(69)	74(74)	75(81)
$\mathcal{H}(I_3)$	75(72)	51(45)	53(45)	66(60)	78(78)	62(64)	69(70)	73(75)	73(76)
DWT(I_1)	78(75)	12(15)	23(19)	50(51)	75(80)	27(32)	35(45)	53(62)	79(80)
LFC(DWT(I_1))$_1$	69(84)	12(14)	18(27)	47(50)	78(70)	32(33)	41(40)	59(60)	80(77)
LFC(DWT(I_1))$_2$	85(83)	56(63)	58(63)	68(74)	82(85)	67(71)	69(75)	76(80)	76(80)
DWT(I_2)	82(80)	53(54)	54(55)	71(68)	85(83)	65(69)	69(73)	77(75)	79(73)
LFC(DWT(I_2))$_1$	80(84)	53(55)	56(56)	69(68)	79(79)	67(68)	71(72)	78(73)	78(73)
LFC(DWT(I_2))$_2$	76(74)	16(19)	22(28)	48(51)	75(74)	32(33)	39(39)	59(57)	73(73)
DWT(I_3)	73(75)	49(50)	49(50)	59(63)	75(75)	67(61)	72(67)	76(76)	79(78)
LFC(DWT(I_3))$_1$	74(80)	50(52)	50(52)	60(63)	73(75)	62(63)	68(69)	73(72)	76(74)
LFC(DWT(I_3))$_2$	73(72)	17(20)	26(30)	51(52)	73(75)	30(23)	40(32)	56(52)	72(68)
KSOFM(I_1)	73(72)	9(7)	13(12)	35(33)	60(56)	32(31)	34(32)	51(50)	65(65)
KSOFM(I_2)	75(74)	17(15)	21(21)	56(55)	85(81)	44(43)	47(46)	67(66)	76(76)
KSOFM(I_3)	66(64)	16(14)	19(20)	47(46)	73(71)	32(31)	36(35)	60(59)	83(82)

(Table 1 continued)

For test set III (Table 1-panel3), a maximum correct target classi cation percentage of 89% for both non-modular and modular network structures is obtained when the input signals $\mathcal{H}(I_1)$ (non-modular network structure) and $\mathcal{F}^a(I_1)$ (modular structure) are used, respectively. In most cases, $\mathcal{F}^a(I_1)$ gives the best range and azimuth estimates. Overall, we can conclude that the networks are fairly robust to variations in target shape, size and roughness.

As an across-the-board conclusion, we may state that the fractional Fourier transform of I_1 with optimal order and low-frequency part of the wavelet transform of I_1 generally represent the best pre-processing options and offer substantial improvements with respect to no pre-processing.

	% of	% of correct r estimation				% of correct θ estimation			
	correct	error tolerance ϵ_r				error tolerance ϵ_θ			
input to ANN	classif.	±0.125 cm	±1 cm	± 5 cm	±10 cm	±0.25°	±2°	±10°	±20°
KSOFM(I_1)	81-81-78	33-21-20	37-27-23	61-55-50	85-79-74	75-65-46	76-68-46	88-88-68	94-91-77
KSOFM(I_2)	85-85-77	41-26-28	44-30-30	71-59-58	90-84-80	80-65-47	82-68-48	93-88-63	97-88-76
KSOFM(I_3)	73-73-67	42-28-28	45-34-30	69-60-59	86-78-81	64-58-44	67-63-46	85-81-69	94-84-84

Table 2. Average percentages of correct classi cation, range (r) and azimuth (θ) estimation for KSOFM used prior to a linear classi er for the three test sets (I-II-III).

The results obtained with KSOFM used prior to linear classi ers are given in Table 2. This combination results in better classi cation performance than when KSOFM is employed prior to ANNs (last three rows of Table 1). The classi cation and azimuth estimation performances are comparable to those obtained with the corresponding unprocessed signals (rst three rows of Table 1). However, range estimation results are inferior compared to the results obtained with unprocessed signals. In any event, this approach is overshadowed by the best pre-processing methods in Table 1.

For networks trained with the GS algorithm, the resulting average percentages of correct type classi cation over all target types are given in Table 3. (Recall that this approach cannot produce localization results.) The maximum average percentage of correct classi cation is 97–98% for both test sets I and II, and can be obtained with either of the input signals $\mathcal{F}(I_1)$, LFC($\mathcal{F}(I_1)$),| LFC($\mathcal{F}(I_1)$)|, $\mathcal{F}^a(I_1)$, $\mathcal{H}(I_1)$, LFC(DWT(I_1))$_1$, or LFC(DWT(I_1))$_2$. It is 91–92% for test set III which can be obtained with either of the input signals $\mathcal{F}(I_1)$, $\mathcal{F}^a(I_1)$, or $\mathcal{H}(I_1)$. We see that the fractional Fourier and low-frequency wavelet transforms again give the best results, though several pre-processing alternatives also give comparable results in this case. Use of KSFOM results in exceptionally poor target differentiation. While the GS algorithm does not offer an advantage over the BP algorithm for test set I, it does offer better results for test set II; the classi cation results obtained with test set II are almost always as good as those with test set I with the GS algorithm, which means that it is accomplishing a very good task of spatial interpolation.

A 100% correct differentiation is achieved with the non-modular ANN trained with the BP algorithm employing DFRT pre-processing. Better range and/or azimuth accuracy can be obtained with some of the other pre-processing methods at the cost of slightly poorer differentiation accuracy. In general, which method is best depends on the relative importance we attach to minimizing errors in differentiation, range, and azimuth. Nevertheless, a compromise which balances both differentiation and localization is obtained with DWT pre-processing using modular networks trained with the BP algorithm and offers 99% differentiation accuracy, 80% or 91% range estimation accuracy and 92% or 98% azimuth estimation accuracy for $\epsilon_r = 0.125$ and 5 cm and $\epsilon_\theta = 0.25°$ and 10°, respectively.

5. CONCLUSION

In this study, different pre-processing methods, structures, and training algorithms for ANNs have been implemented and compared, among which the method leading to the best results emerged. The performance of all the pre-processing methods considered have been compared for three different test sets. The rst test set is based on targets situated at the training locations. The second is based on targets situated at arbitrary locations; it has been observed that ANNs are able to achieve considerable spatial interpolation. The third is based on

input to ANN	% of correct diff.
I_1	95-95-89
I_2	90-90-78
I_3	76-76-68
$\mathcal{F}(I_1)$	97-97-92
LFC($\mathcal{F}(I_1)$)	98-98-86
|LFC($\mathcal{F}(I_1)$)|	97-97-84
$\mathcal{F}(I_2)$	95-95-82
LFC($\mathcal{F}(I_2)$)	96-96-81
|LFC($\mathcal{F}(I_2)$)|	94-94-75
$\mathcal{F}(I_3)$	83-83-69
LFC($\mathcal{F}(I_3)$)	88-88-75
|LFC($\mathcal{F}(I_3)$)|	83-83-71
$\mathcal{F}^a(I_1)$	97-97-91
$\mathcal{F}^a(I_2)$	96-96-83
$\mathcal{F}^a(I_3)$	84-83-71
$\mathcal{H}(I_1)$	97-97-91
$\mathcal{H}(I_2)$	95-95-81
$\mathcal{H}(I_3)$	83-83-71
DWT(I_1)	95-95-89
LFC(DWT(I_1))$_1$	97-97-89
LFC(DWT(I_1))$_2$	97-97-88
DWT(I_2)	91-91-80
LFC(DWT(I_2))$_1$	90-90-79
LFC(DWT(I_2))$_2$	90-90-79
DWT(I_3)	75-75-67
LFC(DWT(I_3))$_1$	77-77-68
LFC(DWT(I_3))$_2$	80-80-71
KSOFM(I_1)	5-8-5
KSOFM(I_2)	13-11-9
KSOFM(I_3)	8-5-6

Table 3. The percentages of correct classi cation for ANNs trained with the GS algorithm for the three test sets (I-II-III).

targets which are not used for training and are somewhat different in size, shape, or roughness than those used for training; it has been observed that the methods are fairly robust in identifying these modi ed targets.

In terms of the number of targets that can be differentiated, correct differentiation percentage, and correct range and azimuth estimation, the use of modular ANNs trained with the BP algorithm, usually with DFRT or DWT pre-processing, gives the best results. With the best optimized methods, it is possible to obtain near perfect differentiation, around 85% correct range estimation and around 95% correct azimuth estimation, which would be satisfactory in a wide range of applications. While the GS algorithm does not offer an advantage over the BP algorithm for test set I, it does offer better results for test set II; the classi cation results obtained with test set II are almost always as good as that with test set I, which means that the GS algorithm is accomplishing a very good task of spatial interpolation.

There is scope for further application of neural networks to sonar, based on the facts that sonar data is difficult to interpret, that the physical models involved can be complex even for simple TOF sonar, and expressions for sonar returns are very complicated even for the simplest target types. Acoustic propagation is also subject to distortion with changes in environmental conditions.

Given the attractive performance-for-cost of sonar-based systems, we believe that the results of this study will be of great usefulness for engineers designing or implementing sonar systems and researchers investigating algorithms and performance evaluation of such systems. While we have concentrated on sonar sensing, the

techniques evaluated and compared in this paper may be useful for other sensing modalities and environments where information from a multiplicity of partial viewpoints must be combined and resolved.

REFERENCES

1. A. Elfes, "Sonar based real-world mapping and navigation," *IEEE Trans. Robotic. Autom.* **RA-3**, pp. 249–265, June 1987.
2. R. Kuc and B. V. Viard, "A physically-based navigation strategy for sonar-guided vehicles," *Int. J. Robotics Res.* **10**, pp. 75–87, April 1991.
3. R. Kuc and M. W. Siegel, "Physically-based simulation model for acoustic sensor robot navigation," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-9**, pp. 766–778, November 1987.
4. A. Kurz, "Constructing maps for mobile robot navigation based on ultrasonic range data," *IEEE Trans. Syst. Man Cy. B* **26**, pp. 233–242, April 1996.
5. J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Navigation*, Kluwer Academic Press, London, U.K., 1992.
6. Ö. Bozma and R. Kuc, "A physical model-based analysis of heterogeneous environments using sonar—ENDURA method," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, pp. 497–506, May 1994.
7. R. Kuc, "Three-dimensional tracking using qualitative bionic sonar," *Robotics Autonomous Syst.* **11**(2), pp. 213–219, 1993.
8. J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE Trans. Robotic. Autom.* **RA-4**, pp. 213–218, April 1988.
9. H. Peremans, K. Audenaert, and J. M. Van Campenhout, "A high-resolution sensor based on tri-aural perception," *IEEE Trans. Robotic. Autom.* **9**, pp. 36–48, February 1993.
10. L. Kleeman and R. Kuc, "Mobile robot sonar for target localization and classi cation," *Int. J. Robotics Res.* **14**, pp. 295–318, August 1995.
11. B. Barshan and B. Ayrulu, "Performance comparison of four time-of- ight estimation methods for sonar signals," *Electron. Lett.* **34**, pp. 1616–1617, 6 August 1998.
12. B. Ayrulu and B. Barshan, "Neural networks for improved target differentiation and localization with sonar," *Neural Networks* **14**, pp. 355–373, April 2001.
13. B. Ayrulu and B. Barshan, "Reliability measure assignment to sonar for robust target differentiation," *Pattern Recogn.* **35**, pp. 1403–1419, June 2002.
14. B. Ayrulu and B. Barshan, "Identi cation of target primitives with multiple decision-making sonars using evidential reasoning," *Int. J. Robotics Res.* **17**, pp. 598–623, June 1998.
15. B. Barshan, B. Ayrulu, and S. W. Utete, "Neural network-based target differentiation using sonar for robotics applications," *IEEE Trans. Robotic. Autom.* **16**, pp. 435–442, August 2000.
16. B. Barshan, "Fast processing techniques for accurate ultrasonic range measurements," *Meas. Sci. Technol.* **11**, pp. 45–50, January 2000.
17. B. Barshan and R. Kuc, "Differentiating sonar re ections from corners and planes by employing an intelligent sensor," *IEEE Trans. Pattern Anal. Mach. Intell.* **12**, pp. 560–569, June 1990.
18. Panasonic Corporation, "Ultrasonic ceramic microphones," 12 Blanchard Road, Burlington, MA, 1989.
19. B. C. Bai and N. H. Farhat, "Learning networks for extrapolation and radar target identi cation," *Neural Networks* **5**, pp. 507–529, May/June 1992.
20. M. Cohen, H. Franco, N. Morgan, D. Rumelhart, and V. Abrash, "Context-dependent multiple distribution phonetic modelling with MLPs," in *Advances in Neural Information Processing Systems*, S. J. Hanson, J. D. Cowan, and C. L. Giles, eds., pp. 649–657, Morgan Kaufmann, San Mateo, CA, 1993.
21. K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamic systems containing neural networks," *IEEE Trans. Neural Networ.* **2**, pp. 252–262, March 1991.
22. M. I. Jordan and R. A. Jacobs, "Learning to control an unstable system with forward modeling," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, ed., pp. 324–331, Morgan Kaufmann, San Mateo, CA, 1990.

23. M. Galicki, H. Witte, J. Dörschel, M. Eiselt, and G. Griessbach, "Common optimization of adaptive processing units and a neural network during the learning period: application in EEG pattern recognition," *Neural Networks* **10**, pp. 1153–1163, August 1997.
24. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, ed., pp. 396–404, Morgan Kaufmann, San Mateo, CA, 1990.
25. E. Alpaydın, "Multiple networks for function learning," in *Proceedings of IEEE Int. Conf. Neural Networks*, pp. 9–14, San Francisco, March 1993.
26. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagation errors," *Nature* **323**, pp. 533–536, 1986.
27. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New Jersey, 1994.
28. Y. Q. Chen, D. W. Thomas, and M. S. Nixon, "Generating-shrinking algorithm for learning arbitrary classi cation," *Neural Networks* **7**(9), pp. 1477–1489, 1994.
29. H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform with Applications in Optics and Signal Processing*, John Wiley & Sons, New York, 2001.
30. D. Mendlovic and H. M. Ozaktas, "Fourier transforms and their optical implementation: I," *J. Opt. Soc. Am. A* **10**, pp. 1875–1881, September 1993.
31. H. M. Ozaktas and D. Mendlovic, "Fourier transforms of fractional order and their optical interpretation," *Opt. Commun.* **101**, pp. 163–169, 15 August 1993.
32. H. M. Ozaktas and D. Mendlovic, "Fractional Fourier transforms and their optical implementation: II," *J. Opt. Soc. Am. A* **10**, pp. 2522–2531, December 1993.
33. H. M. Ozaktas and D. Mendlovic, "Fractional Fourier optics," *J. Opt. Soc. Am. A* **12**, pp. 743–751, April 1995.
34. H. M. Ozaktas, B. Barshan, and D. Mendlovic, "Convolution and ltering in fractional Fourier domains," *Opt. Rev.* **1**, pp. 15–16, November 1994.
35. H. M. Ozaktas, B. Barshan, D. Mendlovic, and L. Onural, "Convolution, ltering, and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transforms," *J. Opt. Soc. Am. A* **11**, pp. 547–559, February 1994.
36. B. Barshan, M. A. Kutay, and H. M. Ozaktas, "Optimal ltering with linear canonical transformations," *Opt. Commun.* **135**, pp. 32–36, February 1997.
37. İ. Ş. Yetik, H. M. Ozaktas, B. Barshan, and L. Onural, "Perspective projections in the space-frequency plane and fractional Fourier transforms," *J. Opt. Soc. Am. A* **17**, pp. 2382–2390, December 2000.
38. Ç. Candan, M. A. Kutay, and H. M. Ozaktas, "The discrete fractional Fourier transform," *IEEE Trans. Signal Proces.* **48**, pp. 1329–1337, May 2000.
39. R. N. Bracewell, *The Hartley Transform*, Oxford University Press, New York, 1986.
40. R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987.
41. D. F. Marshall, W. K. Jenkins, and J. J. Murphy, "The use of orthogonal transforms for improving performance of adaptive lters," *IEEE Trans. Circuits Syst.* **36**, pp. 474–484, April 1989.
42. C. K. Chui, *An Introduction to Wavelets*, Academic Press, San Diego, CA, 1992.
43. S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, pp. 674–693, July 1989.
44. T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.* **43**(1), pp. 59–69, 1982.
45. B. Barshan and B. Ayrulu, "Fractional Fourier transform pre-processing for neural networks and its application to object recognition," *Neural Networks* **15**, pp. 131–140, January 2002.